

1. 如何阅读本章

所有LPC111x系列UART模块都相同。 $\overline{\text{DSR}}$ 、 $\overline{\text{DCD}}$ 和 $\overline{\text{RI}}$ 三个调制信号只在LQFP48和PLCC44封装下配置有引脚。

2. 功能特性

- 16字节收发FIFO。
- 寄存器位置符合' 550工业标准。
- 接收器FIFO触发点可为1、4、8和14字节。
- 内置波特率发生器。
- UART允许软件或硬件实现流控制。
- 支持RS-485/EIA-485 9位模式。
- 调制解调器控制。

3. 引脚描述

表 113. UART引脚描述

引脚	类型	描述
RXD	输入	串行输入。串行接收数据。
TXD	输出	串行输出。串行发送数据。
RTS	输出	请求发送. RS-485 方向控制引脚。
DTR	输出	数据终端就绪。
$\overline{\text{DSR}}$ ^[1]	输入	数据设备就绪。
$\overline{\text{CTS}}$	输入	清除发送。
$\overline{\text{DCD}}$ ^[1]	输入	数据载波检测。
$\overline{\text{RI}}$ ^[1]	输入	振铃指示

[1] 只针对LQFP48封装。

调制解调器输入 $\overline{\text{DSR}}$ 、 $\overline{\text{DCD}}$ 和 $\overline{\text{RI}}$ 复用在两个不同的引脚上。通过使用IOCON_LOC寄存器(见 [7-4.2节](#)) 可为每个功能选择LQFP48封装上引脚的位置, 通过IOCON 寄存器选择功能。

$\overline{\text{DTR}}$ 输出对于2个引脚位置都是可用的。引脚 $\overline{\text{DTR}}$ 的值同时受两个位置分别驱动, $\overline{\text{DTR}}$ 功能在哪个引脚上可以通过IOCON寄存器为引脚选择功能来确定。

4. 时钟和功耗控制

UART模块由AHBCLKCTRL寄存器(见表 3-19)开关。外设UART使用的时钟由波特率发生器产生,由UARTCLKDIV 寄存器所控制(见 表 3-21)。

UART_PCLK时钟信号能通过UARTCLKDIV寄存器来禁止(见第3-4.16节),同时为了降低功耗还可以通过系统AHB时钟控制寄存器的第12位来关闭UART模块(见第3-4.14章节)。

注释: 在UART时钟被允许之前,必需在相应的IOCON寄存器对UART引脚进行配置。

5. 寄存器描述

UART所包含的寄存器如表 9-116组织。除数锁存访问位(DLAB)位于U0LCR[7],它用于允许对除数锁存器的访问。

表 114. 寄存器概览: UART (基地址: 0x4000 8000)

名称	访问类型	地址偏移	描述	复位值 ^[1]	注
U0RBR	只读	0x000	接收器缓存寄存器，存放着下一个要读的已接受字节。	NA	当 DLAB=0
U0THR	只写	0x000	发送保持寄存器。存放下一个将要被发送的字符。	NA	当 DLAB=0
U0DLL	读/写	0x000	除数锁存LSB。波特率除数值的最低字节。完整除数用于分数分频器，来生成波特率。	0x01	当 DLAB=1
U0DLM	读/写	0x004	除数锁存MSB。波特率除数值的最高字节。完整除数用于分数分频器，来生成波特率。	0x00	当 DLAB=1
U0IER	读/写	0x004	中断允许寄存器。包含有7个独立的潜在的UART中断的允许位。	0x00	当 DLAB=0
U0IIR	只读	0x008	中断ID寄存器。识别出哪个中断正被挂起。	0x01	-
U0FCR	只写	0x008	FIFO控制寄存器。控制UART FIFO的用法和模式。	0x00	-
U0LCR	读/写	0x00C	线控制寄存器。包含有帧格式和间断生成的控制。	0x00	-
U0MCR	读/写	0x010	Modem控制寄存器	0x00	-
U0LSR	只读	0x014	线状态寄存器。包含接收和发送状态，以及线错误的标志。	0x60	-
U0MSR	只读	0x018	Modem状态寄存器。	0x00	-
U0SCR	读/写	0x01C	暂存寄存器。软件可以使用这8位临时存储空间。	0x00	-
U0ACR	读/写	0x020	自动波特率控制寄存器。包含自动波特率特性的控制。	0x00	-
-	-	0x024	保留	-	-
U0FDR	读/写	0x028	分数分频寄存器。为波特率分频器产生时钟输入。	0x10	-
-	-	0x02C	保留	-	-
U0TER	读/写	0x030	发送允许寄存器。软件流控制时，使用该寄存器关掉UART发送器。	0x80	-
-	-	0x034 - 0x048	保留	-	-
U0RS485CTRL	读/写	0x04C	RS-485/EIA-485控制。包含配置RS-485/EIA-485模式各个方面的控制。	0x00	-
U0ADRMATCH	读/写	0x050	RS-485/EIA-485地址匹配。包含RS-485/EIA-485模式下的地址匹配值。	0x00	-
U0RS485DLY	读/写	0x054	RS-485/EIA-485直接控制延迟。	0x00	-
U0FIFOLVL	只读	0x058	FIFO等级寄存器。提供当前接收发送FIFO的等级数。	0x00	-

[1] 复位值只反映在所使用位的数据。它不影响保留位内容。

5.1 UART接收缓存寄存器 (U0RBR - 0x4000 8000, 当DLAB = 0, 只读)

U0RBR是UART接收FIFO队列最高字节。接收FIFO队列包含了最早接收到的字符，可通过总线接口读出。LSB（位0）代表最早接收到的数据位。如果接收到的字符小于8位，未使用的MSB填充为0。

如果要访问U0RBR，U0LCR的除数锁存访问位（DLAB）必须为0。U0RBR为只读寄存器。

由于PE、FE和BI位(见表 9-126)与RBR FIFO顶端的字节相对应（即下次读RBR时，读出的字节），因此，将接收的字节及其状态位成对读出的正确方法是先读U0LSR，再读U0RBR。

表 115. UART 接收器缓存寄存器(U0RBR – 地址 0x4000 8000 当 DLAB = 0, 只读) 位域描述

位	符号	描述	复位值
7:0	RBR	UART接收器缓存寄存器包含UART Rx FIFO当中最早接收到的字节。	未定义
31:8	-	保留	-

5.2 UART 发送器保持寄存器（U0THR - 0x4000 8000，DLAB=0，只写）

U0THR是 UART TX(发送) FIFO 的最高字节。该最高字节是TX FIFO中最新的字符，可通过总线接口写入。LSB代表最先要发送的位。

如果要访问U0THR，U0LCR的除数锁存访问位（DLAB）必须为0。U0THR为只写寄存器。

表 116. UART发送器保持寄存器(U0THR – 地址 0x4000 8000 当DLAB = 0, 只写)位域描述

位	符号	描述	复位值
7:0	THR	写UART发送器保持寄存器将数据保存到UART发送FIFO当中。当字节到达FIFO的最底部并且发送器就绪时，该字节将被发送。	NA
31:8	-	保留	-

5.3 UART 除数锁存LSB和MSB寄存器(U0DLL - 0x4000 8000 和 U0DLM - 0x4000 8004, 当 DLAB = 1)

UART除数锁存是UART波特率发生器的一部分，并且保持使用的值，与分数分频器一起去分频UART_PCLK时钟来产生波特率时钟，波特率时钟是波特率的16倍。U0DLL和U0DLM寄存器一起构成一个16位除数，其中U0DLL包含除数的低8位，U0DLM包含除数的高8位；值0x0000被看作是0x0001，因为除数是不允许为0。当访问UART除数锁存寄存器时，U0LCR中的除数锁存访问位（DLAB）必须为1。如何选择U0DLL和U0DLM正确值，详见第9-5.15节。

表 117. UART除数锁存LSB寄存器(U0DLL – 地址 0x4000 8000 当DLAB = 1)位域描述

位	符号	描述	复位值
7:0	DLLSB	UART除数锁存LSB寄存器与U0DLM寄存器一起决定UART的波特率。	0x01
31:8	-	保留	-

表 118. UART除数锁存MSB寄存器(U0DLM – 地址 0x4000 8004 当DLAB = 1) 位域描述

位	符号	描述	复位值
7:0	DLMSB	UART除数锁存LSB寄存器与U0DLM寄存器一起决定UART的波特率。	0x00
31:8	-	保留	-

5.4 UART 中断允许寄存器(U0IER - 0x4000 8004, 当DLAB = 0)

U0IER用于允许UART四个中断源。

表 119. UART中断允许寄存器 (U0IER – 地址 0x4000 8004 当DLAB = 0) 位域描述

位	符号	值	描述	复位值
0	RBR 中断允许		允许UART接收数据有效中断。它还控制字符接收超时中断。	0
		0	禁止RDA中断。	
		1	允许RDA中断。	
1	THRE 中断允许		允许UART THRE中断。该中断的状态可从U0LSR[5]读出。	0
		0	禁止THRE中断。	
		1	允许THRE中断。	
2	RX 线 中断允许		允许UART Rx线状态中断。该中断的状态可从U0LSR出[4:1]中读出。	0
		0	禁止Rx线状态中断。	
		1	允许Rx线状态中断。	
3	-	-	保留	-
6:4	-		保留, 用户软件不要向保留位写入1从保留位读出的值未被定义。	NA
7	-	-	保留	0
8	ABEOIntEn		允许自动波特率结束中断。	0
		0	禁止自动波特率结束中断。	
		1	允许自动波特率结束中断。	
9	ABTOIntEn		允许自动波特率超时中断。	0
		0	禁止自动波特率超时中断。	
		1	允许自动波特率超时中断。	
31:10	-		保留, 用户软件不要向保留位写入1从保留位读出的值未被定义。	NA

5.5 UART中断标识寄存器(U0IIR - 0x4004 8008, 只读)

U0IIR提供状态码用于指示一个挂起中断的中断源和优先级。在访问U0IIR过程中，中断被冻结。如果在访问U0IIR时产生了中断，该中断被记录，下次U0IIR访问可读出。

表 120. UART中断标识寄存器(U0IIR – 地址 0x4004 8008, 只读)位域描述

位	符号	值	描述	复位值
0	IntStatus		中断状态。U0IIR[0]为低有效。挂起的中断可通过U1IIR[3:1]确定。	1
		0	至少有1个中断被挂起。	
		1	没有挂起的中断。	
3:1	IntId		中断标识。U0IER[3:1]指示对应于UART Rx FIFO的中断。下面未列出的U0IER[3:1]的其它组合都为保留值（100, 101, 111）。	0
		011	1 - 接收线状态（RLS）。	
		010	2a - 接收数据可用（RDA）。	
		110	2b - 字符超时指示（CTI）。	
		001	3 - THRE中断。	
		000	4 - Modem 中断。	
5:4	-		保留, 用户软件不要向保留位写入1从保留位读出的值未被定义。	NA
7:6	FIFO Enable		这些位等效于U0FCR[0]。	0
8	ABEOInt		自动波特率结束中断。 如果自动波特率成功完成且中断被允许，则ABEOInt为1。	0
9	ABTOInt		自动波特率超时中断。如果波特率未超时且中断被允许，则ABTOInt为1。	0
31:10	-		保留, 用户软件不要向保留位写入1从保留位读出的值未被定义。	NA

位U0IIR[9:8]通过自动波特率功能设置，指示超时或自动波特率结束。设置自动波特率控制寄存器的Clear位，将清除自动波特率中断条件。

如果IntStatus位是1并且没有中断挂起则IntId位会是0。如果IntStatus是0，且一个非自动波特率中断被挂起，在这种情况下IntId位标识中断类型，其处理过程如表9-121所示。通过U0IIR[3:0]的状态，中断处理程序可以确定中断产生的原因和如何清除激活的中断。为了清除中断优先级，在退出中断服务程序之前必须读U0IIR。

UART RLS中断（U0IIR[3:1]=011）是最高优先级的中断。任何时候，只要UART Rx输入产生四个错误条件（溢出错误（OE）、奇偶错误（PE）、帧错误（FE）和间隔中断（BI））中的任意一个，该中断标志将被置位。产生该中断的UART Rx错误条件可通过查看U0LSR[4:1]得到。当读取U0LSR时清除该中断。

UART RDA中断（U0IIR[3:1]=010）与CTI中断（U0IIR[3:1]=110）共用第二优先级。当UART Rx FIFO到达U0FCR[7:6]所定义的触发点时，RDA被激活。当UART Rx FIFO的深度低于触发点时，RDA复位。当RDA中断激活时，CPU可读出由触发点所定义的数据块。

CTI中断（U0IIR[3:1]=110）为第二优先级中断。当UART Rx FIFO包含至少1个字符并且在接收3.5到4.5字符的时间内没有发生UART Rx FIFO动作时，该中断置位。UART Rx FIFO的任何动作（读或写UART RSR）都将清除该中断。在接收到的信息不是触发等级值的倍数时，CTI中断将会清空UART RBR。例如，如果一个外设想要发送一个105个字符的信息，触发等级值为10个字符；那么CPU将接收10个RDA中断，完成100个字符的传输；之后CPU将收到1-5个CTI中断（取决于服务程序），完成剩下5个字符的传输。

表 121. UART 中断处理

U0IIR[3:0] 值 [1]	优先级	中断类型	中断源	中断复位
0001	-	无	无	-
0110	最高	RX 线 状态 / 错误	OE[2] 或 PE[2] 或 FE[2] 或 BI[2]	读 U0LSR [2]
0100	第二	RX 数据有效	Rx 数据有效或FIFO达到触发等级(U0FCR0=1)	读U0RBR [3] 或UART FIFO 达到触发等级 读U0RBR [3]
1100	第二	字符超时指示	Rx FIFO包含至少1个字符且在一段时间内无字符输入或移出，该时间的长短取决于FIFO 中的字符数以及在3.5到4.5字符的时间内设置的触发等级值。 实际的时间为： $[(\text{字长度}) \cdot 7 - 2] \cdot 8 + [(\text{触发值} - \text{字符数}) \cdot 8 + 1] \text{ RCLK}$	
0010	第三	THRE	THRE[2]	读U0IIR [4] (如果是中断源) 或THR 写操作

[1] 值: "0000", "0011", "0101", "0111", "1000", "1001", "1010", "1011", "1101", "1110", "1111" 保留。
[2] 详见 9–5.9 节 “UART 线状态寄存器 (U0LSR - 0x4000 8014, 只读)”
[3] 详见 9–5.1 节 “UART 接收缓冲寄存器 (U0RBR - 0x4000 8000, 当 DLAB = 0, 只读)”
[4] 详见 9–5.5 节 “UART 中断标识寄存器 (U0IIR - 0x4004 8008, 只读y)” 和第9–5.2 节 “UART 发送保持寄存器 (U0THR - 0x4000 8000 当 DLAB = 0, 只写)”

UART THRE中断(U0IIR[3:1] = 001)是第3优先级中断。当UART THR FIFO为空并且满足特定的初始化条件时，该中断激活。这些初始化条件将使UART THR FIFO被数据填充，以免在系统启动时产生许多THRE中断。在THRE=1时，初始化条件实现了一个字符的延

时减去停止位；并在上一次THRE=1事件之后，在U0THR中没有至少2个字符。在没有译码和THRE中断服务时，该延迟为CPU提供了将数据写入U0THR的时间。如果当前或曾经在UART THR FIFO中有两个或更多字符，而当前U0THR为空时，THRE中断立即被设置。当U0THR写操作或U0IIR读操作发生，且THRE为最高优先级中断（U0IIR[3:1]=001）时，THRE中断复位。

5.6 UART FIFO 控制寄存器 (U0FCR - 0x4000 8008, 只写)

U0FCR控制UART0 RX和TXFIFO的操作。

表 122. UART FIFO控制寄存器(U0FCR – 地址 0x4000 8008,只写) 位域描述

位	符号	值	描述	复位值
0	FIFO Enable	0	UART FIFO被禁止。在应用中必须不能被使用。	0
		1	高电允许对UART Rx和Tx FIFO以及U0FCR[7:1]的访问。该位必须设置以实现正确的UART操作。该位的任何变化都将使UART FIFO自动清空。	
1	RX FIFO Reset	0	对两个UART FIFO都无影响。	0
		1	写1到 U0FCR[1]将清零 UART Rx FIFO中的所有字节并复位指针逻辑。该位自动清零。	
2	TX FIFO Reset	0	对两个UART FIFO都无影响。	0
		1	写1到 U0FCR[2]将清零 UART Rx FIFO中的所有字节并复位指针逻辑。该位自动清零。	
3	-	-	保留	0
5:4	-	-	保留, 用户软件不要向保留位写入1从保留位读出的值未被定义。	NA
7:6	RX Trigger Level	-	这两个位决定在激活中断之前，接收器UART FIFO必须写入多少个字符。	0
		00	触发点0（1个字符或0x01）。	
		01	触发点1（4个字符或0x04）。	
		10	触发点2（8个字符或0x08）。	
		11	触发点3（14个字符或0x0E）。	
31:8	-	-	保留	-

5.7 UART Modem控制寄存器

U0MCR允许modem的回环模式并控制modem的输出信号。

表 123. UART0 Modem控制寄存器 (U0MCR - 地址 0x4000 8010) 位域描述

位	符号	值	描述	复位值
0	DTR Control		modem输出引脚 $\overline{\text{DTR}}$ 的源，该位在回环模式激活时读出为0。	0
1	RTS Control		modem输出引脚 $\overline{\text{RTS}}$ 的源，该位在回环模式激活时读出为0。	0
3-2	-	NA	保留, 用户软件不要向保留位写入1从保留位读出的值未被定义。	0
4	Loopback Mode Select		modem 回环模式提供了一个执行回环测试的诊断机制。发送器输出的串行数据在内部连接到接收器的串行输入端。输入脚RxD对回环模式无影响，输出脚TxD保持总为标记状态。4个modem输入（CTS, DSR, RI和DCD）与外部断开。从外部来看，modem的输出端（RTS, DTR）无效。在内部，4个modem输出连接到4个modem 输入。这样连接的结果是U0MSR的高4位由U0MCR的低4位驱动，而不是在正常模式下由4个modem输入驱动。这样在回环模式下， 就可通过写U0MCR的低4位来允许modem状态中断的产生。	0
		0	禁止modem回环模式。	
		1	允许modem回环模式。	
5	-	NA	保留, 用户软件不要向保留位写入1从保留位读出的值未被定义。	0
6	RTSen	0	禁止自动RTS 流控制。	0
		1	允许自动RTS 流控制。	
7	CTSen	0	禁止自动CTS 流控制。	0
		1	允许自动CTS 流控制。	

5.7.1 自动流控制

如果自动RTS模式被允许，那么UART的接收器FIFO硬件控制UART的 $\overline{\text{RTS}}$ 输出。如果自动CTS模式被允许，若 $\overline{\text{CTS}}$ 输入信号有效，那么UART的U0TSR硬件将只启动发送。

5.7.1.1 自动RTS（Auto-RTS）

设置CTSen位来允许自动RTS功能。自动RTS数据流控制在U0RBR模块中产生，并已编程的接收FIFO触发等级相链接。如果自动RTS被允许，数据流按以下方式控制：

当接收FIFO电平到达已编程的触发等级时， $\overline{\text{RTS}}$ 无效（变为高）。发送UART可在达到触发等级后发送一个额外的字节（假设发送的UART有另一个字节要发送），因为它可能不知道 $\overline{\text{RTS}}$ 的无效直至它开始发送额外的字节后。一旦接收FIFO到达原来的触发等级，RTS就自动重新有效（变低）。RTS的重新有效讲指示发送的UART继续发送数据。

如果自动RTS模式被禁止，那么RTSen位控制UART的 $\overline{\text{RTS}}$ 输出。如果自动RTS模式被允许，那么硬件控制RTS输出且RTS的实际值将在UART的RTSen位中被复制。只要自动 $\overline{\text{RTS}}$ 被允许，则对于软件RTSen位的值只可读。

例如：假设UART工作于type550模式，在U0FCR中设置触发等级为0x2，那么若自动 $\overline{\text{RTS}}$ 被允许，一旦接收FIFO包含8字节（表 9-122），UART将使 $\overline{\text{RTS}}$ 输出无效。一旦接收FIFO到达原来的触发等级：4个字节，那么 $\overline{\text{RTS}}$ 输出就将重新有效。

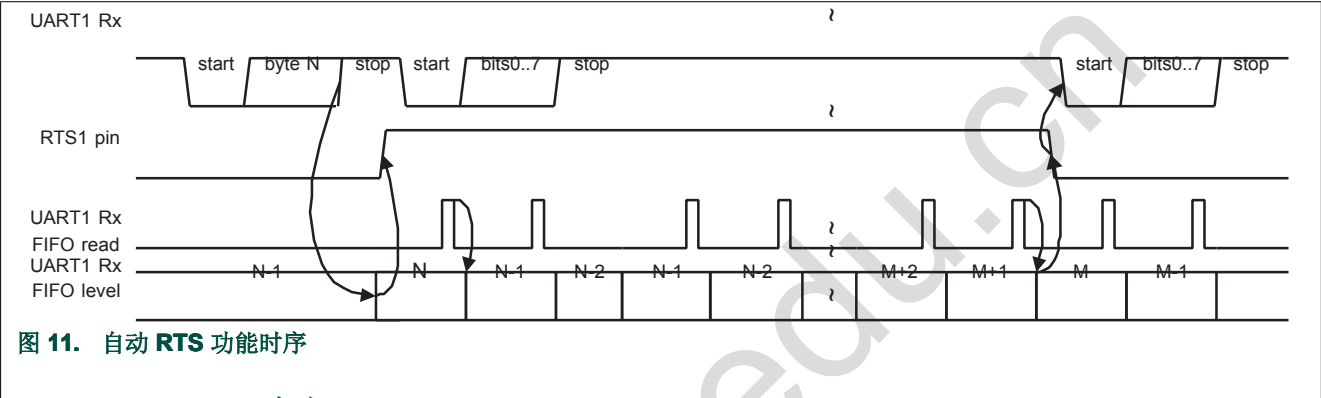


图 11. 自动 RTS 功能时序

5.7.1.2 自动 CTS (Auto-CTS)

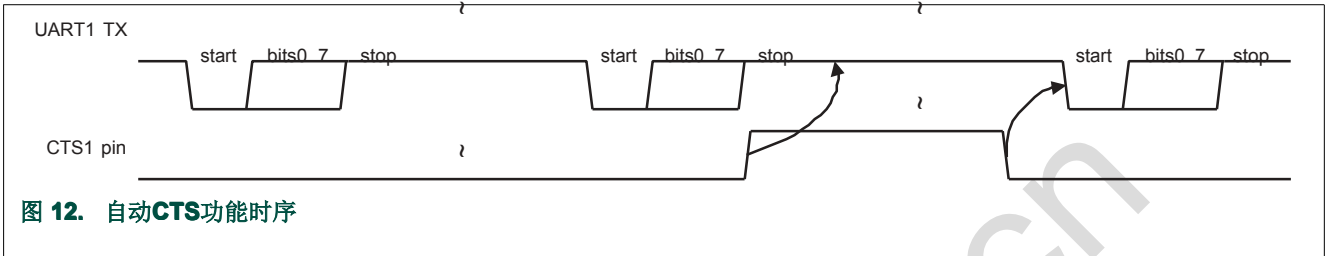
自动CTS功能通过设置CTSen位允许。如果自动CTS被允许，U0TSR模块中的发送器电路在发送下一个数据字节之前检查 $\overline{\text{CTS}}$ 输入。当 $\overline{\text{CTS}}$ 有效（低电平）时，发送器发送下一个字节。为了停止发送器发送后面的字节， $\overline{\text{CTS}}$ 必须在当前发送的最后一个停止位的中间之前被释放。在自动CTS模式中，CTS信号的变化不会触发modem状态中断，除非CTS中断允许位置位，但U0MSR中的Delta CTS位将被置位。表 9-124 列出产生modem状态中断的条件。

表 124. Modem 状态中断产生

允许 modem 状态中断 (U0IER[3])	CTSen (U0MCR[7])	CTS 中断允许 (U0IER[7])	Delta CTS (U0MSR[0])	Delta DCD或后沿RI或 Delta DSR(U0MSR[3]或 U0MSR[2]或U0MSR[1])	Modem 状态中断
0	x	x	x	x	否
1	0	x	0	0	否
1	0	x	1	x	是
1	0	x	x	1	是
1	1	0	x	0	否
1	1	0	x	1	是
1	1	1	0	0	否
1	1	1	1	x	是
1	1	1	x	1	是

此LPC111X 数据手册之中文翻译，由武汉理工大学UP团队友情提供，仅供学习交流之用，欢迎指正，共同修改完善。

自动CTS功能减少到主机系统的中断。当流控制允许时， $\overline{\text{CTS}}$ 状态变化不触发主机中断，因为设备自动控制其自身的发送器。当流控制允许时， CTS 状态变化不触发主机中断，因为设备自动控制其自身的发送器。没有自动CTS，将导致发送器发送任何出现在TX FIFO中的数据，以及接收器溢出错误。图 9-12为自动CTS功能的时序。



字符的发送时， $\overline{\text{CTS}}$ 信号有效。一旦待处理的传输结束，传输就停止。只要 CTS 无效（高电平），UART就继续发送1位。一旦 CTS 无效，传输恢复并发送起始位，后面跟着下一个字符的数据位。

5.8 UART 线控制寄存器 (U0LCR - 0x4000 800C)

U0LCR决定发送和接收数据字符的格式。

表 125. 线控制寄存器 (U0LCR - 地址 0x4000 800C) 位域描述

位	符号	值	描述	复位值
1:0	Word Length Select	00	5位字符长度。	0
		01	6位字符长度。	
		10	7位字符长度。	
		11	8位字符长度。	
2	Stop Bit Select	0	1个停止位。	0
		1	2个停止位(如果U0LCR[1:0]=00则为1.5个)。	
3	Parity Enable	0	禁止奇偶产生和校验。	0
		1	允许奇偶产生和校验。	
5:4	Parity Select	00	奇校验。发送字符和附带的校验位中1的个数为奇数。	0
		01	偶校验。发送字符和附带的校验位中1的个数为偶数。	
		10	校验位强制为“1”	
		11	校验位强制为“0”	
6	Break Control	0	禁止间隔发送。	0
		1	允许间隔发送。当 U0LCR[6]为高电平有效时，输出引脚UART TxD 强制为逻辑0。	

表 125. UART线控制寄存器 (U0LCR - 地址 0x4000 800C) 位域描述

位	符号	值	描述	复位值
7	Divisor Latch Access Bit (DLAB)	0	禁止访问除数锁存	0
		1	允许访问除数锁存	
31: 8	-	-	保留	-

5.9 UART 线状态寄存器 (U0LSR - 0x4000 8014, 只读)

U0LSR为只读寄存器, 提供UART Tx和Rx模块的状态信息。

表 126. UART 线状态寄存器 (U0LSR - 地址 0x4000 8014, 只读) 位域描述

位	符号	值	描述	复位值
0	Receiver Data Ready (RDR)	0	当U0RBR包含未读取的字符时, U0LSR[0]置位; 当UART RBR FIFO为空时, U0LSR[0]清零。	
		0	U0RBR为空。	
		1	U0RBR包含有效数据。	
1	Overrun Error (OE)		溢出错误条件在溢出错误发生后立即设置。读U0LSR操作将清零U0LSR[1]。当UART RSR已经有新的字符组合而UART RBR FIFO已满时, U0LSR[1]置位。此时UART RBR FIFO不会被覆盖, UART RSR中的字符将丢失。	0
		0	溢出错误状态未被激活。	
		1	溢出错误状态被激活。	
2	Parity Error (PE)		当接收字符的校验位为错误状态时产生一个奇偶错误。读U0LSR操作将清零U0LSR[2]位。奇偶错误检测时间取决于U0FCR[0]。 注: 奇偶错误与UART RBR FIFO中顶部的字符相关。	0
		0	奇偶错误状态未被激活。	
		1	奇偶错误状态被激活。	
3	Framing Error (FE)		当接收字符的停止位为0时, 产生帧错误。读U0LSR操作将清零U0LSR[3]。帧错误检测时间取决于U0FCR0。当检测到一个帧错误时, Rx将试图与数据重新同步, 并假设错误的停止位实际是一个超前的起始位。但即使没有出现帧错误, 它也不能假设下一个接收到的字节是正确的。 注: 帧错误与UART RBR FIFO中顶部的字符相关。	0
		0	帧错误状态未被激活。	
		1	帧错误状态被激活。	

表 126. UART 线状态寄存器 (U0LSR - 地址 0x4000 8014, 只读) 位域描述 ...续

位	符号	值	描述	复位值
4	间隔中断 (BI)		在发送整个字符（起始位、数据、校验位和停止位）过程中 Rx D1 如果都保持逻辑0，则产生间隔中断。当检测到间隔条件时，接收器立即进入空闲状态直到 Rx D1 变为全1状态。读 U0LSR 操作将清零该状态位。间隔检测的时间取决于 U0FCR[0]。 注： 间隔中断与 UART RBR FIFO 中顶部的字符相关	0
		0	间隔中断状态未被激活。	
		1	间隔中断状态被激活。	
5	发送器保持寄存器空 (THRE)		当检测到 UART THR 空时，THRE 置位，U0THR 写操作将清零该位。	1
		0	U0THR 包含有效数据。	
		1	U0THR 为空。	
6	发送器空 (TEMT)		当 U0THR 和 U0TSR 为空时，THRE 置位；U0THR 或 U0TSR 包含有效数据将清零该位。	1
		0	U0THR 和/或 U0TSR 包含有效数据。	
		1	U0THR 和 U0TSR 为空。	
7	Rx FIFO 错误 (RXFE)		当一个带有 Rx 错误（例如帧错误、奇偶错误或间隔中断）的字符装入 U0RBR 时，U0LSR[7] 被置位。当读取 U0LSR 寄存器并且 UART FIFO 中不再有错误时，U0LSR[7] 被清零。	0
		0	U0RBR 中没有 UART Rx 错误，或 U0FCR[0]=0。	
		1	UART RBR 包含至少一个 UART Rx 错误 0	
31: 8	-	-	保留	-

5.10 UART Modem 状态寄存器

U0MSR 是一个只读寄存器，它提供 modem 输入信号的状态信息。U0MSR[3:0] 在读取 U0MSR 时被清零。需要注意的是，modem 信号对 UART 的操作没有直接影响。该寄存器用于帮助 Modem 信号的软件实现。

表 127. UART Modem 状态寄存器 (U0MSR - 地址 0x4000 8018) 位域描述

位	符号	值	描述	复位值
0	Delta CTS		当输入 $\overline{\text{CTS}}$ 状态发生变化时，该位置位。读 U0MSR 时清零该位。	0
		0	没有检测到 modem 输入 $\overline{\text{CTS}}$ 上的状态变化。	
		1	检测到 modem 输入 $\overline{\text{CTS}}$ 上的状态变化。	
1	Delta DSR		当输入 $\overline{\text{DSR}}$ 状态发生变化时，该位置位。读 U0MSR 时清零该位。	0
		0	没有检测到 modem 输入 $\overline{\text{DSR}}$ 上的状态变化。	
		1	检测到 modem 输入 $\overline{\text{DSR}}$ 上的状态变化。	

表 127. UART Modem状态寄存器(U0MSR - 地址 0x4000 8018) 位域描述

位	符号	值	描述	复位值
2	后沿 RI		当输入RI发生低到高的跳变时, 该位置位。读取U0MSR时清零。	0
		0	没有检测到modem输入RI上的状态变化	
		1	检测到modem输入RI上低到高的跳变	
3	Delta DCD		当输入DCD的状态发生变化时, 该位置位。读取U1MSR时清零。	0
		0	没有检测到modem输入DCD上的状态变化。	
		1	检测到modem输入DCD上的状态变化。	
4	CTS		清除发送状态。输入信号CTS的补。在回环模式下, 该位连接到U0MCR[1]。	0
5	DSR		数据设备就绪状态。输入信号DSR的补。在回环模式下, 该位连接到U0MCR[0]。	0
6	RI		响铃指示状态。输入信号RI的补。在回环模式下, 该位连接到U0MCR[2]。	0
7	DCD		数据载波检测状态。输入信号DCD的补。在回环模式下, 该位连接到U0MCR[3]。	0

5.11 UART 暂存寄存器 (U0SCR - 0x4000 801C)

在UART操作时U0SCR无效。用户可自由对该寄存器进行读或写。不提供中断接口向主机指示USCR所发生的读或写操作。

表 128. UART暂存寄存器 (U0SCR - 地址 0x4000 8014) 位域描述

位	描述	复位值
7:0 Pad	一个可读可写的字节	0x00
31: - 8	保留	-

5.12 UART 自动波特率控制寄存器 (U0ACR - 0x4000 8020)

UART自动波特率控制寄存器(U0ACR)控制测量波特率生成的输入时钟/数据率的过程, 用户可自由对该寄存器进行读或写。

表 129. 自动波特率控制寄存器 (U0ACR - 地址 0x4000 8020) 位域描述

位	符号	值	描述	复位值
0	Start		自动波特率结束后该位自动清零。	0
		0	自动波特率停止 (自动波特率未运行)。	
		1	自动波特率启动 (自动波特率正在运行)。自动波特率运行位。该位在自动波特率结束后自动清零。	
1	Mode		自动波特率模式选择位。	0
		0	模式 0。	
		1	模式 1。	

表 129. 自动波特率控制寄存器 (U0ACR - 地址 0x4000 8020) 位域描述

位	符号	值	描述	复位值
2	AutoRestart	0	不重新启动	0
		1	如果超时则重新启动（计数器在下一个UART Rx下降沿重新启动）	0
7:3	-	NA	保留，用户软件不要向保留位写入 1。从保留位读出的值未被定义。	0
8	ABEOIntClr		自动波特率中断结束清零位（仅可写访问）。	0
		0	写0无影响。	
		1	写1将在U1IIR中清除相应的中断。	
9	ABTOIntClr		自动波特率超时中断清零位（仅可写访问）。	0
		0	写0无影响。	
		1	写1将在U0IIR中清除相应的中断。	
31:10	-	NA	保留，用户软件不要向保留位写入 1。从保留位读出的值未被定义。	0

5.13 自动波特率

UART自动波特率功能可用于测量基于“AT”协议(Hayes命令)的输入波特率。如果允许，那么自动波特率特性将测量接收数据流的位时，并设置除数锁存寄存器U0DLM和U0DLL。

自动波特率通过设置U0ACR起始位来启动。自动波特率可通过清零U0ACR起始位来停止。一旦自动波特率结束起始位将清零，并且读该位将返回自动波特率的状态（等待/完成）。

可通过U0ACR模式位来选择两种自动波特率测量模式。在模式0中，波特率在UART Rx引脚的两个连续的下降沿上测量（起始位的下降沿和最低位的下降沿）。在模式1中，波特率在UART Rx引脚的下降沿和后续的上升沿之间测量（起始位的长度）

如果超时出现（速率测量计数器溢出），那么U0ACR AutoRestart位可用于自动重新启动速率测量。如果该位置位，速率测量将在UART Rx引脚的下一个下降沿重新启动

自动波特率功能可产生两种中断。

- 如果中断允许，那么将设置U0IIR ABTOInt中断（U0IER ABTOIntEn被置位且自动波特率测量计数器溢出）。
- 如果中断允许，那么将设置U0IIR ABEOInt中断（U0IER ABEOIntEn被置位且自动波特率成功完成

通过设置U0ACR相应的ABTOIntClr和ABEOIntClr位来清除自动波特率中断。

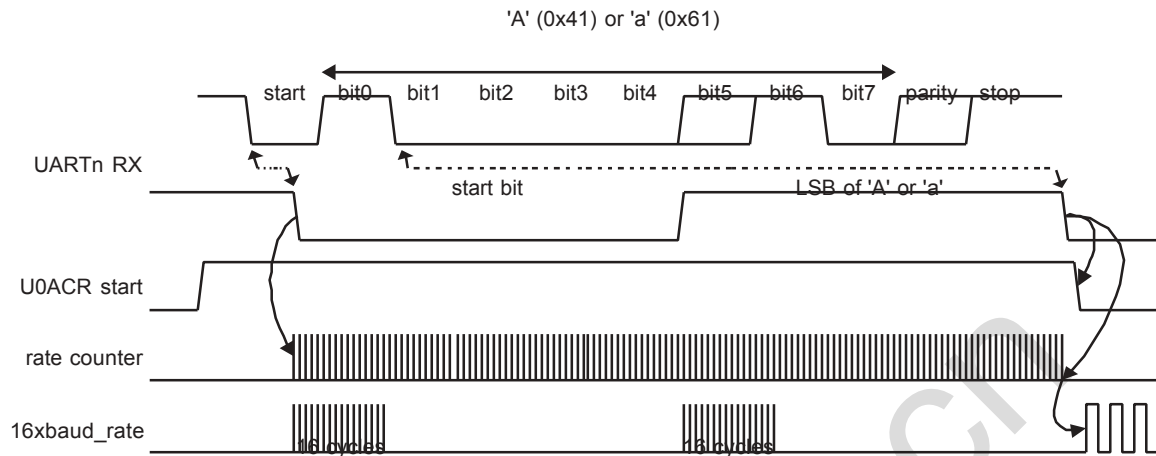
在自动波特率期间, 分数波特率发生器被禁止(DIVADDVAL=0)。也就是, 当自动波特率被使用时, 任何写U0DLM和U0DLL寄存器的操作都应在写U0ACR寄存器前完成。UART0所支持的波特率最小和最大值由UART_PCLK、数据位的个数、停止位和校验位决定。

$$rate_{min} = \frac{2 * PCLK}{16 * 2^{15}} \leq UART_{baudrate} \leq \frac{PCLK}{16 * (2 + \text{数据位} + \text{奇偶位} + \text{停止位})} = rate_{max} \quad (2)$$

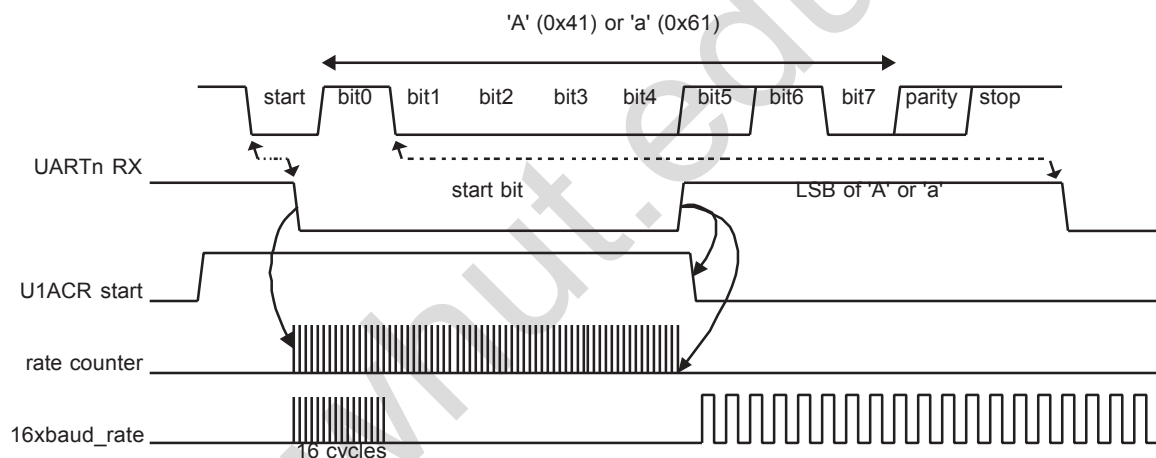
5.14 自动波特率模式

当软件正在期望“AT”命令时, 用期望的字符格式配置UART并设置U0ACR起始位。这里可以不用关心除数锁存器U0DLL和U0DLM的初始值。由于“A”或“a”的ASCII编码是“A”=0x41或“a”=0x61, 因此UART Rx引脚可通过检测起始位以及所期望字符的LSB是两个下降沿来确定AT命令。当U0ACR起始位置位时, 自动波特率协议将执行以下过程

1. 在U0ACR起始位置位时, 速率测量计数器复位且UART U0RSR复位。U0RSR波特率切换为最高的速率。
2. UART Rx引脚下降沿触发起始位的开始。速率测量计数器将开始对PCLK进行计数。
3. 在起始位的接收过程中, RSR的波特率输入端将根据UART输入时钟频率产生16个脉冲, 保证起始位存储在U0RSR中。
4. 在接收起始位(和模式0的字符LSB)的过程中, 速率计数器将按照预分频的UART输入时钟(UART_PCLK)增加。
5. 如果Mode=0, 那么速率计数器将在UART Rx引脚的下一个下降沿停止。如果Mode=1, 那么速率计数器将在UART Rx引脚的下一个上升沿停止。
6. 速率计数器被装入到U0DLM/U0DLL且波特率将自动切换为正常操作模式。设置完U0DLM/U0DLL后, 如果允许, U0IIR ABEOInt(自动波特率结束中断)将被置位。U0RSR将继续接收“A/a”字符剩下的位。



a. 模式0（起始位和最低位用于自动波特率）



b. 模式1（仅起始位用于自动波特率）

图 13. 自动波特率 a) 模式 0、b) 模式 1 波形

5.15 UART 分数分频寄存器 (U0FDR - 0x4000 8028)

UART分数分频寄存器(U0FDR)控制用于产生波特率的时钟预分频器，并可被用户读写。该预分频器根据指定的分数，使用APB时钟来产生一个输出时钟

注意: 如果分数分频被激活(DIVADDVAL > 0)且DLM = 0, DLL寄存器的值必须大于或等于3。

表 130. UART分数分频寄存器(U0FDR - address 0x4000 8028) 位域描述

位	功能	值	描述	复位值
3:0	DIVADDVAL	0	波特率生成预分频除数值。如果该字段为0，分数波特率发生器将不会影响UART波特率。	0
7:4	MULVAL	1	波特率预分频乘数值。不管分数波特率发生器是否使用，该字段必须大于或等于1以使UART正常操作。	1
31:8	-	NA	保留，用户软件不要向保留位写入1。从保留位读出的值未被定义	

该寄存器控制用于波特率生成的时钟预分频器。该寄存器的复位值保持UART禁止时的分数功能，以确保在软件和硬件上与无此特性的UART完全兼容。

可用下式计算UART波特率：

$$UART_{baudrate} = \frac{PCLK}{16 \times (256 \times U0DLM + U0DLL) \times (1 + \frac{DivAddVal}{MulVal})}$$

(3)

其中UART_PCLK为外围时钟，U0DLM和U0DLL为标准UART波特率除数寄存器，DIVADDVAL和MULVAL为UART分数波特率发生器特定参数

MULVAL和DIVADDVAL的值应遵循以下的条件：

1. $1 \leq MULVAL \leq 15$
2. $0 \leq DIVADDVAL \leq 14$
3. $DIVADDVAL < MULVAL$

当正在发送/接收数据或数据时不应修改U0FDR的值，否则数据可能丢失或被破坏。

如果U0FDR寄存器值不遵循这两个请求，那么分数分频输出未定义。如果DIVADDVAL为0，那么分数分频禁止且时钟将不会被分频。

5.15.1 波特率计算

无论是否有分数分频器，UART都可以工作。在现实的应用程序中，可能有若干种分数分频器设置可实现要求的波特率很。以下描述一种设置DLM、DLL、MULVAL和DIVADDVAL值的方法。和要求的值相比，如下设置参数所得到的波特率相对误差小于1.1%。

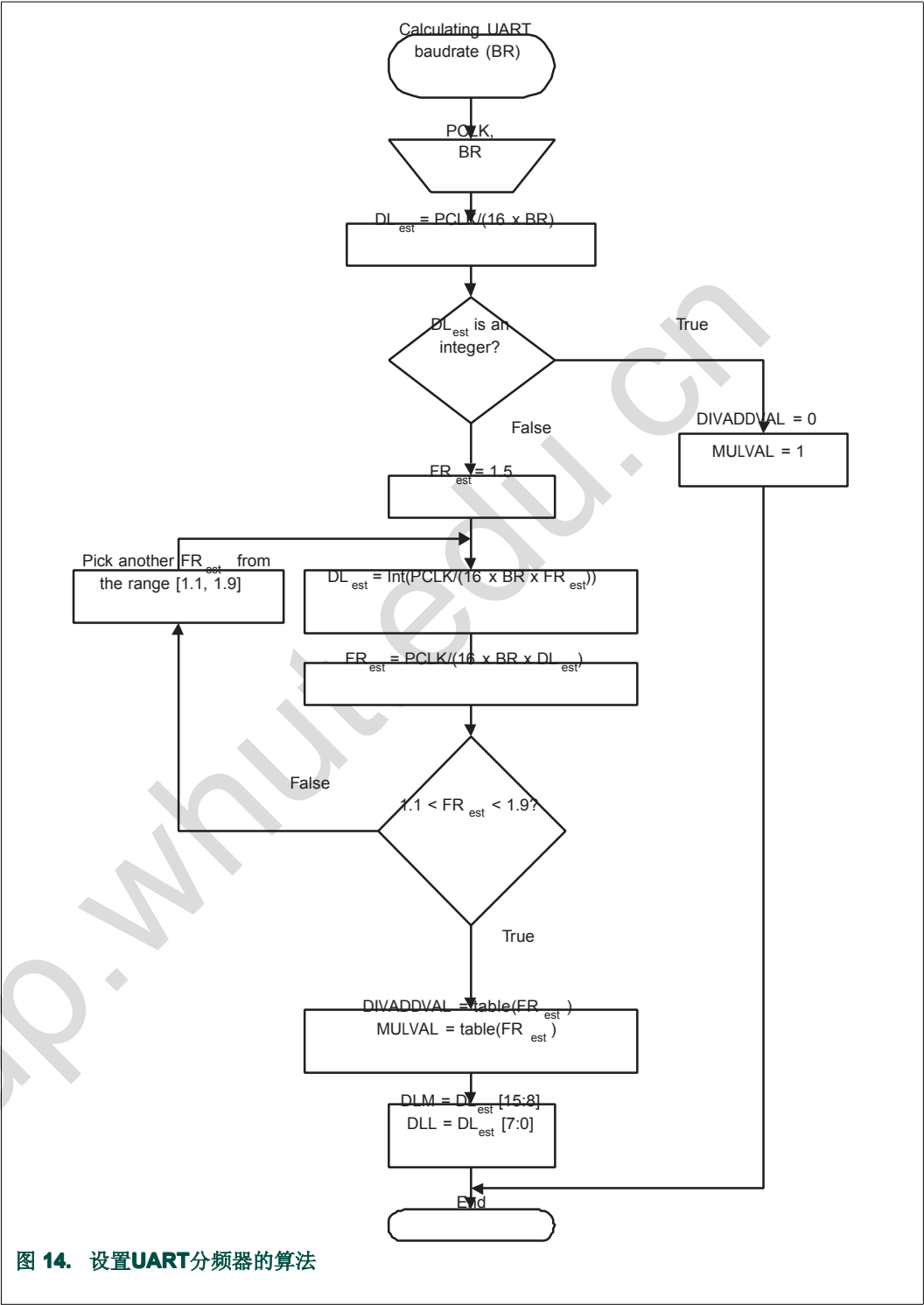


表 131. 分数分频器设置查询表

FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal
1.000	0/1	1.250	1/4	1.500	1/2	1.750	3/4
1.067	1/15	1.267	4/15	1.533	8/15	1.769	10/13
1.071	1/14	1.273	3/11	1.538	7/13	1.778	7/9
1.077	1/13	1.286	2/7	1.545	6/11	1.786	11/14
1.083	1/12	1.300	3/10	1.556	5/9	1.800	4/5
1.091	1/11	1.308	4/13	1.571	4/7	1.818	9/11
1.100	1/10	1.333	1/3	1.583	7/12	1.833	5/6
1.111	1/9	1.357	5/14	1.600	3/5	1.846	11/13
1.125	1/8	1.364	4/11	1.615	8/13	1.857	6/7
1.133	2/15	1.375	3/8	1.625	5/8	1.867	13/15
1.143	1/7	1.385	5/13	1.636	7/11	1.875	7/8
1.154	2/13	1.400	2/5	1.643	9/14	1.889	8/9
1.167	1/6	1.417	5/12	1.667	2/3	1.900	9/10
1.182	2/11	1.429	3/7	1.692	9/13	1.909	10/11
1.200	1/5	1.444	4/9	1.700	7/10	1.917	11/12
1.214	3/14	1.455	5/11	1.714	5/7	1.923	12/13
1.222	2/9	1.462	6/13	1.727	8/11	1.929	13/14
1.231	3/13	1.467	7/15	1.733	11/15	1.933	14/15

5.15.1.1 例 1: UART_PCLK = 14.7456 MHz, BR = 9600

根据提供的算法 $DL_{est} = PCLK / (16 \times BR) = 14.7456 \text{ MHz} / (16 \times 9600) = 96$ 。而这个 DL_{est} 是一个整数, $DIVADDVAL = 0$, $MULVAL = 1$, $DLM = 0$, 且 $DLL = 96$ 。

5.15.1.2 例 2: UART_PCLK = 12 MHz, BR = 115200

根据提供的算法 $DL_{est} = PCLK / (16 \times BR) = 12 \text{ MHz} / (16 \times 115200) = 6.51$ 。这个 DL_{est} 不是一个整数并且下一步要估计FR的参数。使用初始估计 $FR_{est} = 1.5$, 新的 $DL_{est} = 4$ 被计算出来同时 FR_{est} 被计算出来为 $FR_{est} = 1.628$ 。因为 $FR_{est} = 1.628$ 是在1.1到1.9范围之内, $DIVADDVAL$ 和 $MULVAL$ 的值可以查表获取。

在表 9-131 中最接近 $FR_{est} = 1.628$ 的值是 $FR = 1.625$ 。它相当于 $DIVADDVAL = 5$ 和 $MULVAL = 8$ 。

根据这些发现, 建议UART的设置如下: $DLM = 0$ 、 $DLL = 4$ 、 $DIVADDVAL = 5$ 和 $MULVAL = 8$ 。根据 式 9-3, UART的波特率应该是115384。这个波特率和原始指定的115200相对误差为0.16%。

5.16 UART 发送允许寄存器 (U0TER - 0x4000 8030)

除配备了完整的硬件流控制(如前描述的auto-cts和auto-rts机制)之外, 通过U0TER可实现软件流控制。当 $TxE_n = 1$ 时, 如果UART发送器有效将会持续发送数据。当 TxE_n 变成0, 则会停止UART发送。

虽然 表 9-132 描述了如何使用TxEn位实现硬件流，但是还是强烈建议让UART硬件实现自动控制流，并限制TxEn实现软件控制流的范围。

表 9-132 描述了如何使用TXEn位去实现软件控制流。

表 132. UART 发送允许寄存器 (U0TER – 地址 0x4000 8030) 位域描述

位	符号	描述	复位值
6:0	-	保留，用户软件不要向该位写入1。从保留位读出的值未被定义	NA
7	TXEN	该位为1（复位值）时，如果以前的数据都被发送后，写入THR 1 的数据被输出到Tx D引脚。如果一个字符正在发送时该位清零，则结束这个字符的发送，后面的字符也不再发送，直到该位重新被置位。换言之，该位为0将终止THR或Tx FIFO 到发送移位寄存器的字符传输。当检测到硬件握手Tx-允许信号CTS出错或利用软件握手接收到一个XOFF字符（DC3）时，软件将该位清零。当检测到正确的Tx-允许信号或接收到XON 字符（DC1）时，软件又能将该位重新置位。	1
31:8	-	保留	-

5.17 UART RS485 控制寄存器 (U0RS485CTRL - 0x4000 804C)

U0RS485CTRL 寄存器控制UART RS-485/EIA-485模式的配置。

表 133. UART RS485控制寄存器(U0RS485CTRL – 地址0x4000 804C) 位域描述

位	符号	值	描述	复位值
0	NMMEN	0	RS-485/EIA-485普通多点模式(NMM)被禁止。	0
		1	RS-485/EIA-485普通多点模式(NMM)允许。在该模式下，当接收到的字节是地址时，将会导致UART设置校验错误并产生一个中断。	
1	RXDIS	0	接收器被允许。	0
		1	接收器被禁止。	
2	AADEN	0	自动地址检测(AAD)被禁止。	0
		1	自动地址检测(AAD)被允许。	
3	SEL	0	如果方向控制被允许(位DCTRL = 1)，引脚RTS被用于方向控制。	0
		1	如果方向控制被允许(位DCTRL = 1)，引脚DTR被用于方向控制。	
4	DCTRL	0	禁止自动方向控制。	0
		1	允许自动方向控制。	
5	OINV		该位使在引脚RTS (或DTR)上的方向控制极性反转。	0

表 133. UART RS485控制寄存器(U0RS485CTRL – 地址0x4000 804C) 位域描述 ...续

位	符号	值	描述	复位值
		0	当发送器已将数据发送出去时, 方向控制引脚会被置为逻辑 '0'。当数据的最后一位也被发送出去之后它会被置为逻辑 '1'。	
		1	当发送器已将数据发送出去时, 方向控制引脚会被置为逻辑 '1'。当数据的最后一位也被发送出去之后它会被置为逻辑 '0'。	
31:6	-	-	保留, 用户软件不要向保留位写入1。从保留位读出的值未被定义。	NA

5.18 UART RS-485地址匹配寄存器 (U0RS485ADRMATCH - 0x4000 8050)

U0RS485ADRMATCH寄存器包含用于RS-485/EIA-485模式的地址匹配值。

表 134. UART RS-485地址匹配寄存器(U0RS485ADRMATCH – 地址0x4000 8050) 位域描述

位	符号	描述	复位值
7:0	ADRMATCH	包含地址匹配值。	0x00
31:8	-	保留	-

5.19 UART1 RS-485 延迟值寄存器 (U0RS485DLY - 0x4000 8054)

用户可通过对8位RS485DLY寄存器编程来设置: 最后一个停止位离开TXFIFO到 $\overline{\text{RTS}}$ (或 $\overline{\text{DTR}}$)有效之间的延迟。这段延迟是在波特率时钟周期之内的。延迟可编程设置为0-255个位时间。

表 135. UART RS-485 延迟值寄存器(U0RS485DLY – 地址 0x4000 8045)位域描述

位	符号	描述	复位值
7:0	DLY	包含方向控制(RTS或 DTR)延迟值。该寄存器与一个8位计数器连接进行工作。	0x00
31:8	-	保留, 用户软件不要向保留位写入1。从保留位读出的值未被定义。	NA

5.20 RS-485/EIA-485 操作模式

RS-485/EIA-485的特性允许将UART配置成可寻址的从设备。可寻址的从设备是众多受控于一个主设备中的一个

UART主设备发送器通过检查校验位(第九位)是否设置 '1', 来识别地址字符。对于数据字符, 校验位是0

每个UART从设备接收器都能够被分配到一个不同的地址。从设备可以编程设置为人工或者自动的方式拒绝不是它们自己地址的数据。

RS-485/EIA-485普通多点模式 (NMM)

通过将RS485CTRL位设置为0, 来允许该模式。在该模式下, 当一个接收的字符被检测为地址时, 将会导致UART设置校验错误并产生一个中断。

如果接收器禁止(RS485CTRL位1 = '1'), 任何接收到的数据字节都会被忽略且不会存储到RXFIFO。当一个地址字节被检测到了(parity位 = '1'), 它会被放置在RXFIFO同时生成一个Rx数据准备中断。处理器可以读地址字节, 决定是否允许接收器去接受后续数据

当接收器被允许时(RS485CTRL位1 = '0'), 所有接收到的数据都会被接受并存储到RXFIFO, 无论它们是数据或地址。当一个地址字符被接收时, 将会产生一个校验错误中断, 由处理器决定是否禁止接收器。

RS-485/EIA-485 自动地址检测(AAD)模式

当RS485CTRL寄存器位0(9位模式允许)和2(AAD模式允许)都被设置时, UART处在自动地址检测模式。

在该模式, 接收器将会把它接收到的任何地址字节(parity = '1')和RS485ADRMATCH寄存器中的8位值(可编程设置)进行比较。

若接收器被禁止(RS485CTRL位1 = '1'), 如果接收到字节是数据字节或是与RS485ADRMATCH值不匹配的地址字节, 则都会被丢弃。

若检测到一个匹配的地址字符, 它会带着校验位一起被压入RXFIFO中, 同时接收器会自动被允许(RS485CTRL位1会被硬件清除)。接收器也会生成一个Rx数据准备中断。

当接收器被允许(RS485CTRL位1 = '0')的时候, 所有接收的字节都会被接受并存储到RXFIFO, 直到收到一个与RS485ADRMATCH值不匹配的地址字节。当这个情况发生的时候, 接收器会自动被硬件禁止(RS485CTRL位1会被设置), 不匹配的地址字符将不会被存储在RXFIFO中。

RS-485/EIA-485自动方向控制

RS485/EIA-485模式允许发送器自动控制DIR引脚的状态, 它可以作为方向控制的输出信号。

通过设置RS485CTRL位4 = '1'允许这个特性。

如果允许方向控制, 当RS485CTRL位3 = '0'时会使用 $\overline{\text{RTS}}$ 引脚, 在RS485CTRL位3 = '1'时则使用 $\overline{\text{DTR}}$ 引脚。

当允许自动方向控制的时候, 在CPU写数据进入TXFIFO时候被选择的引脚会有效(驱动为低)。当数据最后一位被发送出去之后, 引脚变为无效(驱动为高)。见RS485CTRL寄存器的位4和5。

除了回环模式之外, RS485CTRL位4优于所有其他控制方向引脚的机制。

RS485/EIA-485 驱动器延迟时间

驱动器延迟时间是指最后一个停止位离开TXFIFO到RTS信号失效这一段时间。该段延迟时间可在8位RS485DLY寄存器中进行编程设置。延迟时间是在波特率时钟周期中的一部分，其值可以设置为0-255个位时间。

RS485/EIA-485 输出反转

引脚RTS(或DTR)方向控制信号的极性可以通过对寄存器U0RS485CTRL第5位编程来反转。该位被设置，则当发送器中有数据正在等待被发送时，方向控制引脚会被驱动为逻辑1。当数据的最后一位已被发送出去时，方向控制引脚会被设置为逻辑0。

5.21 UART FIFO 等级寄存器(U0FIFOLVL - 0x4000 8058, 只读)

U0FIFOLVL寄存器是一个只读寄存器，允许软件读出现在FIFO的等级状态。发送和接收FIFO等级都呈现在该寄存器中。

表 136. UART FIFO等级寄存器(U0FIFOLVL – 地址 0x4000 8058, 只读) 位域描述

位	符号	描述	复位值
3:0	RXFIFILVL	反映UART接收器FIFO现在的等级。 0=空，0xF=FIFO满。	0x00
7:4	-	保留。从保留位读出来的值未定义。	NA
11:8	TXFIFOLVL	反映UART发送器FIFO现在的等级。 0=空，0xF=FIFO满。	0x00
31:12	-	保留。从保留位读出来的值未定义。	NA

6. Architecture

UART的结构如下方框图所示。

APB接口提供CPU或主机与UART之间的通信连接。

UART接收器模块U0RX监视串行输入线RxD0的有效输入。UART RX移位寄存器（U0RSR）通过RxD0接受有效的字符。当U0RSR接收到一个有效字符时，它将该字符传送到UART RX 缓存寄存器FIFO中，等待CPU或主机通过主机接口进行访问。

UART发送器模块U0TX接受CPU或主机写入的数据并将数据缓存到UART TX保持寄存器FIFO(U0THR)中。UART TX移位寄存器（U0TSR）读取U0THR中的数据，并将数据通过串行输出引脚TxD0发出。

UART波特率发生器模块U0BRG产生UART TX模块所使用的定时。U0BRG模块时钟源为UART_PCLK。主时钟与U0DLL和U0DLM寄存器所设定的除数分频，得到Tx模块使用的时钟。该时钟为16倍过采样时钟NBAUDOUT。

中断接口包含寄存器U0IER和U0IIR。中断接口接收几个由U0Tx和U0Rx模块发出的单时钟宽度的允许信号。

U0Tx和U0Rx的状态信息保存在U0LSR中。U0Tx和U0Rx的控制信息保存在U0LCR中。

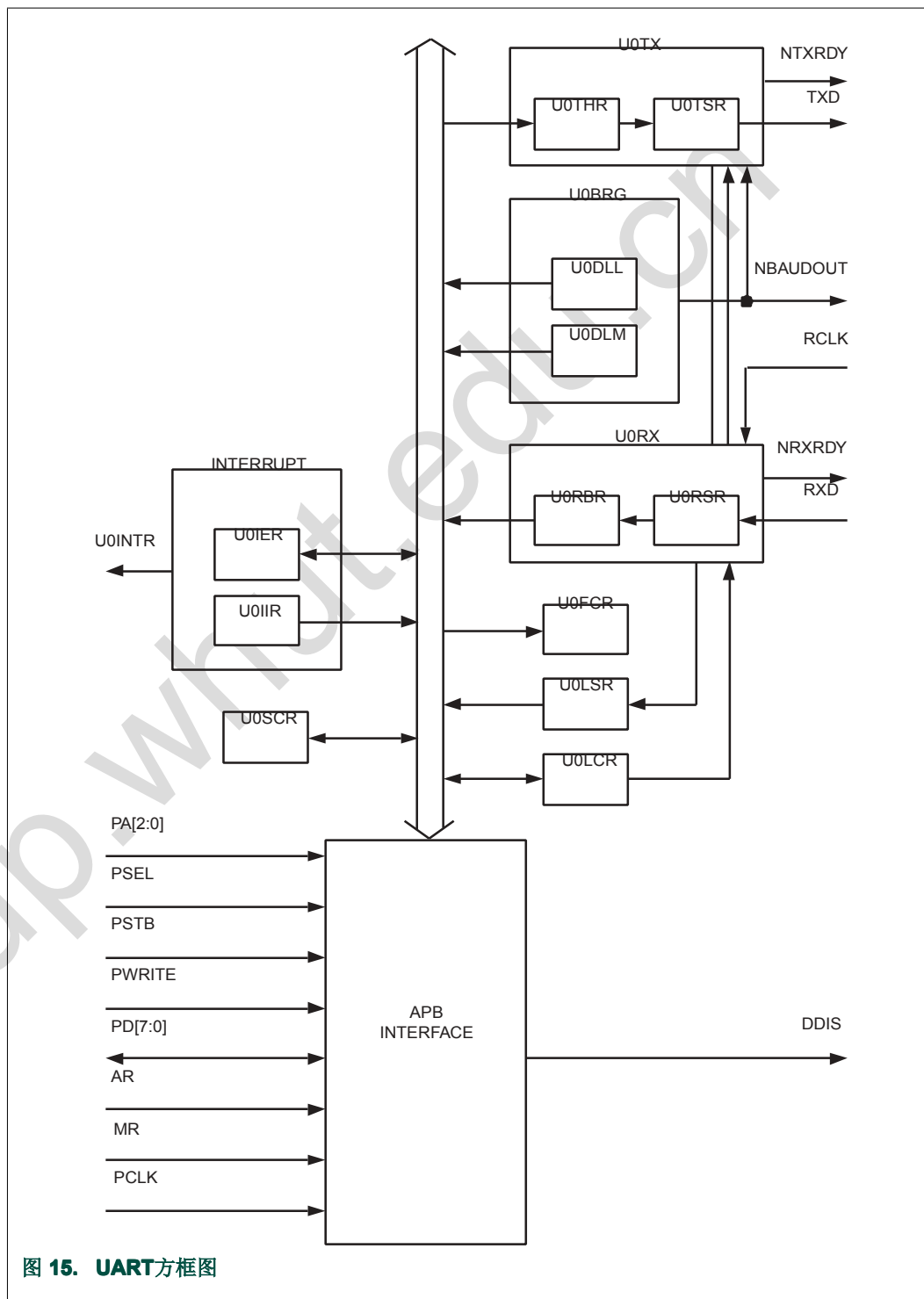


图 15. UART方框图